Floor Acquisition Multiple Access with Collision Resolution

Rodrigo Garcés and J.J. Garcia-Luna-Aceves
Baskin Center for Computer Engineering and Information Sciences
University of California at Santa Cruz
Santa Cruz, CA 95064, USA
garces@cse.ucsc.edu jj@cse.ucsc.edu

Abstract— **Collision avoidance and resolution multiple access (CARMA) protocols are presented and analyzed. These protocols use a floor acquisition multiple access strategy based on carrier sensing, together with collision resolution of floor requests (RTS) based on a tree-splitting algorithm. For analytical purposes, an upper bound is derived for the average costs of resolving collisions of floor requests using the tree-splitting algorithm. This bound is then applied to the computation of the average channel utilization in a fully connected network with a large number of stations. Under light-load conditions, CARMA protocols achieve the same average throughput as floor acquisition multiple access (FAMA) protocols. It is also shown that, as the arrival rate of RTSs increases, the throughput achieved by CARMA protocols is close to the maximum throughput that any FAMA protocol can achieve if propagation delays and the control packets used to acquire the floor are much smaller than the data packet trains sent by stations. Simulation results validate the simplifying approximations made in the analytical model. Our analysis results indicate that collision resolution makes floor acquisition multiple access much more effective.**

I. Introduction

Several medium access control (MAC) protocols have been proposed over the past few years that are based on three- or four-way handshake procedures meant to reduce the number of collisions among data packets, thereby providing better performance than the basic ALOHA or CSMA protocols [2], [3], [4], [5], [6], [7], [9], [10]. The concept of "floor acquisition" was first introduced by Fullmer and Garcia-Luna-Aceves [5] for MAC protocols based on such handshake procedures. In a single-channel network, floor acquisition entails allowing one and only one station at a time to send data packets without collisions. To achieve this, a station that wishes to send one or multiple data packets must send a request-to-send packet (RTS) to an intended destination and receive a clear-to-send packet (CTS) from it, before it is allowed to transmit any data. RTSs are required to last a minimum amount of time that is a function of the channel propagation time. A floor acquisition strategy is very attractive in the control of packet-radio networks, because it provides a building block to solve the hidden-terminal problem that arises in CSMA [11]. Variants of this basic strategy can be designed using different types of contention-based MAC protocols like ALOHA or CSMA to transmit RTSs into the channel, and three- or four-way handshakes can be implemented (i.e., RTS-CTS-DATA or RTS-CTS-DATA-ACK).

This work was supported in part by DARPA under Grant DAAB07-95-C-D157.

Although floor acquisition multiple access (FAMA) protocols are able to sustain higher loads than CSMA [5], their throughput still degrades rapidly once stations start retransmitting unsuccessful RTSs that collide repeatedly with other RTSs. This paper shows that using tree-splitting algorithms to resolve the collision of RTSs in a FAMA protocol running in a high-speed network improves substantially the performance of FAMA protocols under high-load conditions. This is because data packets never collide with control packets in a FAMA protocol and the propagation delays and duration of RTSs and CTSs are much smaller than the duration of data packets, which means that the average time needed to resolve the collisions of RTSs is very small compared to the duration of data packets. Therefore, even if every RTS has to go through collision resolution, the channel can still be used for useful data the majority of the time. In contrast, its is well known that tree-splitting algorithms do not provide much improvement for CSMA protocols [1], which stems from the fact that the data packets themselves are used for collision resolution.

Section II describes a specific protocol, which we call CARMA (for collision avoidance and resolution multiple access), and which uses non-persistent carrier sensing for the transmission of RTSs and a tree-splitting algorithm to resolve collisions of RTSs. Section III computes an upper bound on the average costs of resolving RTS collisions, i.e., the times associated with the eventual successful transmission of all data packets involved in a collision-resolution tree; the importance of these bounds is that they are independent of the number of stations in the network. Section IV uses them to compute a lower bound of average throughput achieved by CARMA when a very large population of nodes is assumed. We show that the throughput achieved by CARMA as the arrival rate of RTSs increases is very close to the maximum throughput that can be achieved by any FAMA protocol when the propagation delays and control packets are much smaller than the data packets or packet trains sent by stations. Simulation results help us validate the simplifications used in our analytical model. Section V offers our concluding remarks.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **1996**	2. REPORT TYPE	3. DATES COVERED **00-00-1996 to 00-00-1996**

4. TITLE AND SUBTITLE **Floor Acquisition Multiple Access with Collision Resolution**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

II. CARMA

A FAMA protocol requires a station who wishes to send one or more packets to acquire the right to use the channel exclusively (called the floor) before transmitting the data packets. In FAMA-NTR [5], before transmitting a data packet, a station senses the state of the channel to see if it is idle or not. If the channel is busy, the station backs off and tries to acquire the channel at a later time; on the other hand, if the channel is sensed to be free, the station sends an RTS. In short, stations follow a non-persistent CSMA strategy for the transmission of RTSs. The sender listens to the channel for one maximum round-trip time plus the time needed for the destination to send a CTS. If the CTS is not corrupted and is received within the time limit, the transmission of data packets from the sender proceeds. The CTS is sent by the destination station to let other stations in the system know that the floor of the channel has been acquired. Accordingly, when a station receives a correct CTS, it backs off until the channel is released by the sender.

Although each station transmits an RTS only when it determines that the channel is free, a collision with other RTS transmissions may still occur due to propagation delays. RTSs are vulnerable to collisions for time periods equal to the propagation delays between senders of RTSs. During these periods, multiple stations may sense the channel free and also send RTSs, thus causing collisions.

FAMA protocols solve collisions by backing off and rescheduling RTS transmissions [5], [6]. As with CSMA protocols, this procedure yields good results if the RTS traffic is low; however, the probability of RTS collisions increases as the rate of RTS transmissions increases, with a corresponding decrease of system throughput. Eventually, as the RTS transmission rate increases, the constant RTS collisions cause the channel to collapse, bringing the flow of data packets to a halt. To remedy this problem, we present CARMA (for collision avoidance and resolution multiple access), which is a new variant of FAMA protocols for fully-connected networks in which all stations can listen to one another.

CARMA uses carrier sensing for the transmission of RTSs and a tree-splitting algorithm to resolve collisions of RTSs. Throughout this paper, we assume a simple tree-splitting algorithm, but more sophisticated collision-resolution algorithms [1] can be used instead. Each station must know the maximum number of stations allowed in the system and the maximum propagation delay in the network. For the slotted version of CARMA, a time slot is assumed to last as long as the maximum propagation delay.

A. Protocol Description

Each station is assigned a unique identifier, a stack and two variables ($LowID$ and $HiID$). $LowID$ is initially the lowest ID number that is allowed to send an RTS, while $HiID$ is the highest ID number that is allowed to send an RTS. Together they constitute the allowable ID interval that can send RTSs, i.e., attempt to acquire the floor. If the ID of a station is not within this interval, it cannot send its RTS. As we describe subsequently, the stack is simply a storage mechanism for ID intervals that are waiting to get permission to send an RTS.

A station can be in one of five different states in CARMA, namely:

- PASSIVE: The station has no local packets pending and no transmissions are detected in the channel.
- RTS: The station is trying to acquire the floor and has sent an RTS.
- XMIT: The station has the floor and is exchanging data packets.
- REMOTE: The station is receiving transmissions from other stations, and started to detect channel activity before it had any local packet to send.
- BACKOFF: The station has local packets pending and had to reschedule its request for the floor.

When a passive station has one or multiple packets to send, it first listens to the channel. If the channel is busy (i.e., carrier is detected), the station backs off and reschedules its RTS at some time into the future. Alternatively, if the channel is clear (i.e., no carrier is detected) for one maximum round-trip time, the station transmits an RTS. The sender then waits and listens to the channel for one maximum round-trip time plus the time needed for the destination to send a CTS. When the originator receives the CTS from the destination, it acquires the floor and begins transmitting its data packet burst. The sender is limited to a maximum number of data packets, after which it must release the channel and must compete for the floor at a later time if it still has data packets to send.

If the sender of an RTS does not receive a CTS within a time limit, the sender as well as all other stations in the system know that a collision has occurred. As soon as the first collision takes place, every station divides the ID interval $(LowID, HiID)$ into two ID intervals. The first ID interval, which we will call the backoff ID interval, is $(LowID, \lceil \frac{HiID+LowID}{2} \rceil - 1)$, while the second ID interval, the allowed ID interval, is $(\lceil \frac{HiID+LowID}{2} \rceil, HiID)$. Each station in the system updates the stack by executing a PUSH stack command, where the key being pushed is the backoff ID interval. After this is done, the station updates $LowID$ and $HiID$ with the values from the allowed ID interval. This procedure is repeated each time a collision is detected.

Only those stations that were in the RTS state at the time the first collision occurred are allowed into the collision-resolution phase of the protocol. All other stations will be in REMOTE state until all collisions are resolved. Collision resolution evolves in terms of collision-resolution intervals. In the first interval of the collision-resolution phase all stations in the allowed ID interval that are in the RTS state try to retransmit an RTS. If none of the stations within this ID interval request the channel, the channel will be idle for a time period equal to a maximum channel delay (τ). At this point, a new update of the stack and of the variables $LowID$ and $HiID$ is due. Each station executes a POP command in the stack. This new ID interval now becomes the new $HiID$ and $LowID$. The same procedure takes place if, during the first collision-resolution interval, only one station is requesting the channel; the originator receives the CTS from the destination and begins transmitting its data packet burst, after which the station releases the channel and transitions to the PASSIVE state. The total time for this successful transmission is the duration of an RTS, a CTS, the data packet burst, plus three channel delays. The third alternative is for multiple stations to request the channel causing a collision. The stations in the allowed ID interval are once more split into two new ID intervals and the stack as well as the variables for each station are updated. In this case, the duration of the collision-resolution interval is equal to the collision time plus the channel delay. The algorithm repeats these steps until all the RTS collision have been resolved. Notice that as soon as the backoff stack becomes empty and there are no values in the allowable interval, all stations know that all the collisions have been resolved. Accordingly, once the tree-splitting algorithm terminates, all stations are either in the PASSIVE state, or in the BACKOFF state if they have packets to send but were not in the RTS state at the beginning of the first collision that started the tree-splitting algorithm. A waiting period of two times the maximum channel delay during which the channel is idle occurs upon termination of the tree-splitting algorithm. The next access to the channel is driven by the arrival of new packets to the stations and the transmission of RTSs that have been backed off.

To permit the transmission of packet bursts, CARMA enforces waiting periods on receiving stations at strategic points in the operation of the protocol. A station that has received a data packet in the clear must wait for one maximum propagation time after processing a data packet, this allows the sender to send more packets if desired. A station that has understood any control packet must wait for twice the duration of the maximum propagation time; this allows correct RTS-CTS exchanges to take place. On the other hand, if a transmitting station is in the RTS state, the protocol enforces a waiting period of two maximum propagation times after sending its RTS. This allows the destination to receive the RTS and transmit the corresponding CTS. A sending station must also wait one maximum propagation time after the last data packet of its packet train.

B. Example

We now illustrate CARMA using a simple example assuming that time is slotted. Each station has a distinct position in the leaves of a binary tree based on its ID. If n is the total number of stations in the system, the binary tree has $2 \times n + 1$ nodes. The root of the tree is labeled as n_r and its right and left child as n_1 and n_0, respectively. For each of the other nodes, the labels are composed of the parent label, plus a 0 if it is the left child or a 1 if it is a right child. As an example, take a system with four stations labeled n_{00}, n_{01}, n_{10}, and n_{11}. The binary tree has a total of seven nodes with the four stations as its leaves. The root of the tree has the label n_r. The left child of the root node is n_1 while its right child is n_0. Station n_0 is the parent node of n_{01} and n_{00}. Similarly, station n_{10} is the right child of node n_1, while station n_{11} is its left child. We define the subtree T_{label} as the subtree at node n_{label}. In our example, the subtree for node n_{01} is T_{01}.

Assume that, at time t_o, we are at node n_r and we are allowed to listen simultaneously at all the stations of its subtree T_r for a time period of τ seconds. Only one of the following three things can occur:

•**Case 1–Idle:** There are no RTSs in any of the leaves (stations) in subtree T_r; therefore, the channel is idle. This lasts τ seconds.

•**Case 2–Success:** There is only one RTS in the subtree T_r; therefore, there is no collision and a station acquires the floor. This lasts one successful transmission period T_s.

•**Case 3–Collision:** There are two or more stations (leaves) in the subtree T_r sending an RTS; therefore, a collision occurs. This lasts one failed transmission period T_f.

Assume that, at time t_0, Case 3 occurs with station n_{00} and n_{01} each sending an RTS in the same slot, while station n_{10} and station n_{11} do not request the channel. Fig. 1 illustrates this. The first collision occurs at time t_0; all stations in the system notice the beginning of the resolution algorithm and update their stacks and their $LowID$ as well as their $HiID$ values. Stations n_{00} and n_{01} are members of the backoff ID interval; therefore, they wait until the collisions in the allowed ID interval are resolved. They both are excluded from sending RTSs. After a time period T_f, Stations n_{10} and n_{11} are allowed to request for the channel. Since stations n_{10} and n_{11} in tree T_1 do not wish the channel, the first case applies here. After τ seconds, all stations notice that the channel is idle, which means that there were no collisions in tree T_1. All the stations in the system must update their intervals and the stack. They execute a POP-stack command and the new allowable interval is (n_{00}, n_{01}); therefore, T_0 can proceed to solve its RTS collisions. Both stations n_{00} and n_{01} transmit an RTS control packet and Case 3 occurs again. Since a collision occurred, the interval is split, i.e.,, the subtree T_0 is split in two halves, T_{00} and T_{01}. Station n_{01} is within the allowable interval while the n_{00} station must wait, its interval is the top of the stack. Since T_{01} has only one station requesting the channel, that station acquires the floor and transmits its data package. After T_s seconds, the stations do an update, a POP command in the stack and n_{00} can request and acquire the channel transmitting its data package. At this point, all the stations know that all the collisions have been resolved, because the stacks are empty and there are no valid values in the allowable interval.

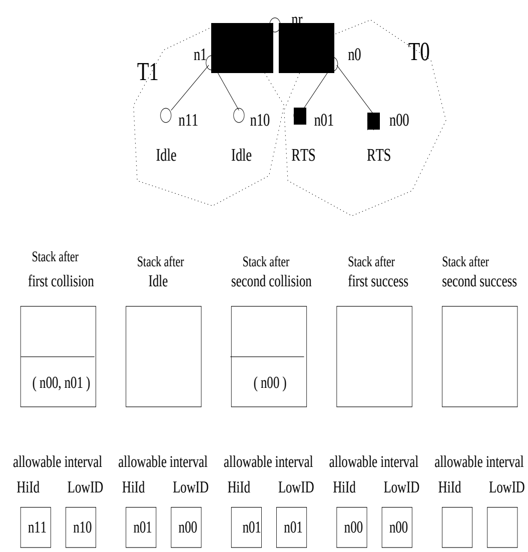

Fig. 1. Tree structure to solve the collisions for a system with $n = 4$ and $m = 2$. $\mathcal{C}(4, 2) = 2$, $\mathcal{S}(4, 2) = 2$ and $\mathcal{Z}(4, 2) = 1$.

III. Average Collision Resolution Costs

In this section, we present upper bounds for the average costs of resolving RTS collisions. Every station can listen to the transmissions of any other station. For the purpose of our analysis, we assume that: (a) the channel introduces no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly, (b) two or more transmissions that overlap in time in the channel must all be retransmitted, (c) a packet propagates to all stations in exactly τ seconds [8], (d) each station has at most one data block to send at any time and transmits the entire data block when it acquires the floor, and (e) time is slotted in τ-second slots for the slotted version. The average size of a data block is δ seconds, and RTS and CTS packets last γ seconds. Both δ and γ are assumed to be multiples of τ.

A. Average Cost Analysis

The binary tree is a structure defined on a finite set of nodes composed of three disjoint sets: a root node, a binary tree called a left subtree and a binary tree called a right subtree. As we have described, there are only three possible cases to consider for the resolution of RTS collisions: idle, success, or collision. For each of these cases, we wish to find a recursive equation expressing its average cost, i.e., the number of subtrees starting from the root that need to be visited before all the stations with an RTS to send have been serviced. To do so, we consider a system with n stations and m RTSs arriving during a contention time period τ. Because each station in the system is assigned one or no RTS at any given time, a leaf of the binary tree, is assigned an "RTS" or an "idle," depending on whether or not it has an RTS to send. We assign three distinct average cost values: $\overline{\mathcal{Z}}(n,m)$ for the idle case, $\overline{\mathcal{S}}(n,m)$ for the success case, and $\overline{\mathcal{C}}(n,m)$ for the collision case. These three costs depend on the number of leaves n and the number of stations with RTSs, m. They represent an average number over all the possible permutations of m RTSs in n total stations. What each of these costs actually means and what rules apply to each of the three cases can be explained by means of a simple example.

The number of permutations of a tree with four leaves ($n = 4$), given that two out of the four stations are requesting the channel simultaneously ($m = 2$), is six. In our prior example, stations n_{00} and n_{01} each sends an RTS in the same time slot, while station n_{10} and station n_{11} remain idle. This tree represents just one of the six possible permutations. Let us assign the index i to to the ith permutation and calculate $\mathcal{S}_i(4,2)$ and $\mathcal{C}_i(4,2)$, respectively. Starting with the root node, the first thing that happens is a collision because two stations are competing for the channel. Following the rules of the algorithm, we take subtree T_0 and once again a collision occurs. The total collision cost is $\mathcal{C}_i = 2$. The next step is to go to node n_{00} and transmit the data packet, followed by the transmission of the data packet in node n_{01}. The total successful cost is $\mathcal{S}_i = 2$. The final step is to take subtree T_1. Because no RTSs are present, the total zero cost is $\mathcal{Z}_i = 1$. What we have counted is the number of subtrees that have collisions ($\mathcal{C}_i$), the number of subtrees that only have one RTS in them ($\mathcal{S}_i$) as well as the number of subtrees that are idle ($\mathcal{Z}_i$). The same counting procedure can be repeated for each of the $\binom{4}{2} = 6$ permutations of trees with $n = 4$ and $m = 2$. The six possible combinations contribute equally to the total average costs $\overline{\mathcal{C}}(4,2)$, $\overline{\mathcal{S}}(4,2)$ and $\overline{\mathcal{Z}}(4,2)$. In general, the average for each of the three types of cost can be calculated by adding each individual permutation cost and by dividing by the total number of permutations.

For counting purposes, a subtree that has no RTS stations or only one RTS station does not have to be explored further down. Counting can stop there and one unit can be added to either $\mathcal{Z}$ or $\mathcal{S}$. It is interesting to observe that $\mathcal{S}$ is always equal to m. Based on this example, we can deduce the general rules shown in Table I.

Rule 1:	$\mathcal{C}$(n,0) = 0	Rule 4:	$\mathcal{S}$(n,0) = 0	Rule 7:	$\mathcal{Z}$(n,0) = 1
Rule 2:	$\mathcal{C}$(n,1) = 0	Rule 5:	$\mathcal{S}$(n,m) = m	Rule 8:	$\mathcal{Z}$(n,1) = 0
Rule 3:	$\mathcal{C}$(n,n) = n-1	Rule 6:	$\mathcal{S}$(n,n) = n	Rule 9:	$\mathcal{Z}$(n,n) = 0

TABLE I
AVERAGE COST RULES.

A.1 Average Collision Cost

In our example of the binary tree with four leaves ($n = 4$) and station n_{00} and n_{01} each sending an RTS while station n_{10} and station n_{11} remain idle, we found the collision cost to be $\mathcal{C}_i(4,2) = 2$ units. This tree can be visualized as two disjoint binary subtrees and the parent node. Similarly, the total collision cost can be expressed as the cost of the right subtree, plus the cost of the left subtree plus one unit cost for the root of the tree. This result can be extended to the general case, yielding the following equation:

$$c_{root\ tree} = c_{right\ subtree} + c_{left\ subtree} + 1 \tag{1}$$

In addition to our example, there are five other possible ways to distribute two stations with an RTS to send in four positions. Table II shows all six cases and the collision cost $\mathcal{C}_i$ associated with each of them.

n_{11}	n_{10}	n_{01}	n_{00}	C Cost	S Cost	Z Cost
idle	idle	RTS	RTS	2	2	1
idle	RTS	idle	RTS	1	2	0
idle	RTS	RTS	idle	1	2	0
RTS	idle	idle	RTS	1	2	0
RTS	idle	RTS	idle	1	2	0
RTS	RTS	idle	idle	2	2	1

TABLE II
COST TABLE FOR ALL POSSIBLE PERMUTATIONS FOR A TREE WITH $n = 4$ AND $m = 2$.

To calculate the average cost equation for $\overline{\mathcal{C}}(4,2)$, we sum each individual cost and divide this result by six. Notice that we have four cases in which the right and left subtrees each has one RTS station. There is one case in which the left subtree has two RTS stations and one case in which the right subtree has two RTS stations. The average cost $\overline{\mathcal{C}}(4,2)$ can be expressed in the following compact equation:

$$\overline{\mathcal{C}}(4,2) = \sum_{i=0}^{2} \frac{\binom{2}{2-i}\binom{2}{i}}{\binom{4}{2}} \left[\overline{\mathcal{C}}(2, 2-i) + \overline{\mathcal{C}}(2,i) + 1 \right] \tag{2}$$

The recursion splits the original tree for $n = 4$ into two subtrees, each for $n = 2$, plus the root node. This result can be extended to the average collision cost $\overline{\mathcal{C}}(n,m)$. The average number of subtrees satisfying this requirement for a binary tree with n leaves and m RTS-requests is

$$\overline{\mathcal{C}}(n, m) = \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \left[\overline{\mathcal{C}}(\alpha, m - i) + \overline{\mathcal{C}}(\beta, i) + 1\right] \tag{3}$$

where

$$\begin{aligned}
\alpha &= \lceil n/2 \rceil;\ \beta = n - \alpha = n - \lceil n/2 \rceil \\
\mu &= \begin{cases} 0 & \text{if } m \leq \alpha \\ m - \alpha & \text{if } m > \alpha \end{cases} \\
\nu &= \begin{cases} m & \text{if } m \leq \beta \\ \beta & \text{if } m > \beta \end{cases}
\end{aligned}$$

If n is even, $\alpha = \beta = \frac{n}{2}$; otherwise, $\beta = \alpha - 1$. There are three possible μ-ν combinations. First, if $m \leq \alpha$ and $m \leq \beta$, then $\mu = 0$ and $\nu = m$. In the second case, $m \leq \alpha$ and $\beta < m$; therefore, $\mu = 0$, while $\nu = \beta$. Finally, if m is greater than both α and β, then $\mu = m - \alpha$ and $\nu = \beta$. Note that the parameter m cannot be $> \alpha$ and $\leq \beta$ at the same time because $\beta \leq \alpha$; accordingly, this case is excluded.

Each cost unit for $\overline{\mathcal{C}}(n, m)$ has a time overhead of T_f and contributes negatively to the overall throughput of the system. The next average cost, $\overline{\mathcal{S}}(n, m)$, contributes positively to the overall throughput of the system and is very simple to calculate.

A.2 Average Success Cost

Any tree or subtree, regardless of its size, will have a success cost of one unit if there exist only one station with an RTS to send. This is clearly the case since we need to visit the root node of such a tree or subtree and stop there. In the case of a tree with m stations with an RTS to send, there are exactly m subtrees with one RTS each; otherwise, they would be a subtree with no RTS nodes or more than two RTSs creating a collision. For both of this cases the success cost is $\mathcal{S} = 0$. Table II shows the success cost table for our example with the six permutations. As Table II shows, the cost is always 2. The total average success cost for any tree of size n with m RTS stations is simply Rule 6 from Table I, i.e., $\overline{\mathcal{S}}(n, m) = m$. Each success cost unit has a time period of $T_s = 2\gamma + 3\tau + \delta$. The success cost increases the average throughput of the system. Each time an RTS is successful a station can transmit one or more data packets.

A.3 Average Idle Cost

According to Rule 8 in Table I, no matter how big the tree or subtree is, if there is only one RTS leaf ($m = 1$), the idle cost will always be zero.

In our example of the binary tree with four leaves ($n = 4$) and two stations with an RTS to send ($m = 2$), we can plot a cost table with all the six permutation cases, as shown in Table II. The idle cost at the root node can be expressed as the cost for the right subtree, plus the left subtree. This result can be extended to the general case, yielding the following equation:

$$\mathcal{Z}_{root\ tree} = \mathcal{Z}_{right\ subtree} + \mathcal{Z}_{left\ subtree} \tag{4}$$

To calculate the average cost for our example, we sum each of the individual costs and divide this result by six permutations. There are four cases in which the right and left subtrees each has one station with an RTS to send. There is one case in which the left subtree has two stations with an RTS to send, and one case in which the right subtree has two stations with an RTS to send. The average cost $\overline{\mathcal{Z}}(4, 2)$ can be expressed in the following equation:

$$\overline{\mathcal{Z}}(4, 2) = \sum_{i=0}^{2} \frac{\binom{2}{2-i}\binom{2}{i}}{\binom{4}{2}} \left[\overline{\mathcal{Z}}(2, 2 - i) + \overline{\mathcal{Z}}(2, i)\right] \tag{5}$$

This result can be extended to the average idle cost for any n leaves and m stations with an RTS to send, $\overline{\mathcal{Z}}(n, m)$.

$$\overline{\mathcal{Z}}(n, m) = \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \left[\mathcal{Z}(\alpha, m - i) + \mathcal{Z}(\beta, i)\right] \tag{6}$$

The parameters α, β, μ and ν are the same as in Eq. (4). The time period for each cost unit for the idle cost case is equal to the channel delay τ. There exists a dependence among each of the three costs. For any tree with n leaves and m stations requesting the channel, the following equation is true:

$$\mathcal{S}(n, m) + \mathcal{Z}(n, m) - \mathcal{C}(n, m) - 1 = 0 \tag{7}$$

B. Upper Bounds

The next question that we address is computing upper bounds for all three average costs of collision resolution. Obviously, the upper bound of the average success cost $\overline{\mathcal{S}}(n, m)$ is equal to m. The following two theorems provide the upper bound for the average idle cost $\overline{\mathcal{Z}}(n, m)$ and the upper bound for the average collision cost $\overline{\mathcal{C}}(n, m)$.

Theorem 1: For all $m > 1$ and $n > 1$, $\overline{\mathcal{Z}}(n, m) \leq \frac{m}{2}$.

Theorem 2: For all $m > 1$, $\overline{\mathcal{C}}(n, m) \leq \frac{3m}{2} - 1$.

The proofs of these theorems are in the Appendix. The proofs use mathematical induction. From Eq. (4) we know that there are three possible μ-ν combinations, determining the indices of the summation. First, if $m \leq \alpha$ and $m \leq \beta$, then $\mu = 0$ and $\nu = m$. In the second case, $m \leq \alpha$ and $\beta > m$; therefore, $\mu = 0$ while $\nu = \beta$. Finally, if m greater then both α and β, then $\mu = m - \alpha$ and $\nu = \beta$. Again, we note that the parameter m cannot be $> \alpha$ and $\leq \beta$ at the same time since $\beta \leq \alpha$, and we disregard this case.

IV. Throughput Analysis

The analysis in this section makes the same assumptions introduced in the previous section and uses the same traffic model used for the FAMA-NTR protocol [5]. Given that the upper bounds on average collision-resolution costs are independent of the number of stations, we approximate the traffic into the channel with an infinite number of stations, each having at most one RTS to send at any time, and forming a Poisson source sending RTSs with an aggregate mean generation rate of λ packets per unit time. With this model, the average number of RTS arrivals in a time interval of length T is λT, i.e., $m = \lambda T$. All data blocks have a duration of δ seconds. The average channel throughput is given by

$$s = \frac{\overline{U}}{\overline{B} + \overline{I}} \quad (8)$$

where $\overline{U}$ is the average utilization time of the channel, during which the channel is being used to transmit data packets; $\overline{B}$ is the expected duration of a busy period, during which the channel is busy with successful or unsuccessful transmissions; and $\overline{I}$ is the average idle period, i.e., the average interval between two consecutive busy periods.

A. Unslotted CARMA

A successful transmission consists of an RTS with one propagation delay to the intended recipient, a CTS with a propagation delay to the sender, and a data packet followed by a propagation delay. Therefore, the average duration period for a successful transmission is

$$T_s = 2\gamma + 3\tau + \delta. \quad (9)$$

For an RTS to be successful, it must be the only packet in the channel during its transmission. Its probability of success equals the probability that no arrivals occur in τ seconds, because there is a delay across the channel of τ seconds before all the other stations in the network detect the carrier signal. After this vulnerability period of τ seconds, all stations defer their transmissions. Therefore, given that arrivals of RTSs to the channel are Poisson with parameter λ, we obtain

$$P_s = P\{\text{ No arrivals in } \tau \text{ seconds}\} = e^{-\lambda\tau} \quad (10)$$

The number of stations that participate in the collision-resolution phase is $m = \lambda\tau$. Within the tree, the three cases of the collision resolution discussed in the previous sections are present. Each one of them has an average upper bound cost that is independent of the number of stations (n), but is a function of the number of stations requesting the channel (m). The question now is: what are the time periods associated with each of this cases? In the case of a colliding transmission $(m > 1)$, the time period consists of one RTS package followed by one or more RTSs transmitted by other stations within time Y, where $0 \leq Y \leq \tau$, plus one propagation delay τ. The average failed transmission period is bounded by [5]:

$$T_f \leq \gamma + 2\tau \quad (11)$$

As in the FAMA-NTR protocol, a waiting period of τ seconds is required after a successful transmission, while a failed transmission requires a waiting period of 2τ seconds. A busy period is composed of both the successful and the tree transmission periods. The duration of an average busy period equals the sum of the percentage of successful transmission periods times their duration, T_s, plus the percentage of the tree periods times their duration. The tree periods are composed of three parts, corresponding to success, idle, and collision periods, each with a distinct cost and duration. According to the upper bounds derived in section III-B, the average busy period can be bounded as follows:

$$\overline{B} \leq T_s \cdot P_s + \left(T_f(\frac{3m}{2} - 1) + \tau\frac{m}{2} + T_s m\right)(1 - P_s) \quad (12)$$

Substituting the values for P_s, T_f, T_s and m, we obtain

$$\begin{aligned} \overline{B} \quad \leq \quad & \left(3\gamma + 5\tau + \delta - \frac{7}{2}\gamma\lambda\tau - \frac{13}{2}\lambda\tau^2 - \delta\lambda\tau\right) \cdot e^{-\lambda\tau} + \\ & \left(\frac{7}{2}\gamma\lambda\tau - \gamma + \frac{13}{2}\lambda\tau^2 - 2\tau + \delta\lambda\tau\right) \end{aligned} \quad (13)$$

The channel carries user data for δ seconds each time an RTS is sent successfully without collision resolution, and δ seconds for each of the RTSs that collide when collision resolution is applied. Therefore, the average utilization is:

$$\overline{U} = \delta \cdot P_s + \delta m(1 - P_s) = (1 - \lambda\tau)\delta \cdot e^{-\lambda\tau} + \delta\lambda\tau \quad (14)$$

The average idle period is equal to the average interarrival time plus the average waiting period enforced. It is the same as in the unslotted FAMA-NTR protocol [5].

$$\begin{aligned} \overline{I} &= \frac{1}{\lambda} + \tau \cdot P_s + 2\tau \cdot (1 - P_s) \\ &= \frac{1}{\lambda} + \tau \cdot e^{-\lambda\tau} + 2\tau \cdot (1 - e^{-\lambda\tau}) \end{aligned} \quad (15)$$

Substituting Eqs. (12), (14) and (15) into Eq. (8) a lower bound of average throughput of CARMA is given by

$$S \quad \geq \quad \frac{2\delta\lambda e^{-\lambda\tau}(\lambda\tau - 1) - 2\delta\lambda^2\tau}{A \cdot e^{-\lambda\tau} + B} \quad (16)$$

where

$$\begin{aligned} A &= -6\lambda\gamma - 8\lambda\tau - 2\lambda\delta + 7\gamma\lambda^2\tau + 13\lambda^2\tau^2 + 2\lambda^2\tau\delta \\ B &= -7\gamma\lambda^2\tau + 2\gamma\lambda - 13\lambda^2\tau^2 - 2\lambda^2\tau\delta - 2 \end{aligned}$$

B. Slotted CARMA

In this section, we use the same assumptions used for unslotted CARMA. The channel is slotted and each slot lasts as long as the maximum propagation delay τ. With slotting, stations are restricted to start transmissions only on slot boundaries.

As it was the case in unslotted CARMA, the average duration period for a successful transmission is given by Eq. (9). The probability that an RTS is successful is

$$P_s = P\{k = 1 \text{ arrival in a slot}|\text{some arrivals in a slot}\} = \frac{\lambda\tau \cdot e^{-\lambda\tau}}{1 - e^{-\lambda\tau}} \quad (17)$$

In the case of a colliding transmission $(m > 1)$, the time period consists of one RTS followed by a propagation delay τ. All colliding RTSs are sent at the beginning of the same slot; accordingly, we have

$$T_f = \gamma + \tau \quad (18)$$

As it was done for unslotted CARMA, $\overline{B}$ can be bounded according to Eq (12). Substituting the values for P_s, T_f, T_s and m, we obtain

$$\begin{aligned} \overline{B} \quad \leq \quad & T_s \cdot P_s + \left(T_f(\frac{3m}{2} - 1) + \tau\frac{m}{2} + T_s m\right)(1 - P_s) \\ = \quad & \frac{7\gamma\lambda\tau + 10\lambda\tau^2 + 2\delta\lambda\tau - 2\gamma - 2\tau}{2(1 - e^{-\lambda\tau})} + \\ & \frac{(-7\gamma\lambda^2\tau^2 - 10\lambda^2\tau^3 - 2\delta\lambda^2\tau^2 + 2\gamma\lambda\tau - 2\lambda\tau^2 + 2\gamma + 2\tau) \cdot e^{-\lambda\tau}}{2(1 - e^{-\lambda\tau})} \end{aligned} \quad (19)$$

The average utilization is:

$$\overline{U} = \delta \cdot P_s + \delta m(1 - P_s) = \frac{\delta \lambda \tau (1 - \lambda \tau \cdot e^{-\lambda \tau})}{(1 - e^{-\lambda \tau})} \tag{20}$$

The average idle period is the same as in the slotted FAMA-NTR protocol [5], i.e.,

$$\begin{aligned} \overline{I} &= \tau \cdot \frac{1}{1 - e^{-\lambda \tau}} + \tau \cdot P_s + 2\tau \cdot (1 - P_s) \\ &= \frac{3\tau - (\lambda \tau^2 + 2\tau) \cdot e^{-\lambda \tau}}{(1 - e^{-\lambda \tau})} \end{aligned} \tag{21}$$

Substituting Eqs. (19), (20),and (21) into Eq. (8) a lower bound on the average throughput of slotted CARMA can be written as follows:

$$S \geq \frac{2\delta \lambda^2 \tau^2 e^{-\lambda \tau} - 2\delta \lambda \tau}{A \cdot e^{-\lambda \tau} + B} \tag{22}$$

where

$$\begin{aligned} A &= (\gamma \lambda \tau + 4\lambda \tau^2 + 10\lambda^2 \tau^3 + 2\tau + 7\gamma \lambda^2 \tau^2 - 2\gamma + 2\lambda^2 \tau^2 \delta) \\ B &= (-10\lambda \tau^2 - 4\tau - 7\gamma \lambda \tau + 2\gamma - 2\delta \lambda \tau) \end{aligned}$$

C. Numerical Results

We compare CARMA with FAMA-NTR for the cases of a low-speed network (9600 b/s) and high-speed network (1Mb/s) in which either small data packets (53 bytes) or large data packets (400 bytes) are transmitted. We assume the distance between stations to be the same and define the diameter of the network to be 1 mile. Assuming these parameters, the propagation delay of the channel is 5.4μs. In order to accommodate the use of IP addresses for destination and source, the minimum size of RTSs and CTSs is 20 bytes. We normalize the throughput results by setting $\delta = 1$ and defining the following variables

$a = \frac{\tau}{\delta}$	(normalized propagation delay)
$b = \frac{\gamma}{\delta}$	(normalized control packets)
$G = \lambda \cdot \delta$	(offered load, normalized to data packet)

TABLE III
NORMALIZED VARIABLES

Substituting the new normalized variables from Table III into Eq. (16), we obtain

$$S \geq \frac{2e^{-aG}(aG - 1) - 2aG}{A' \cdot e^{-aG} + B} \tag{23}$$

where

$$\begin{aligned} A' &= -6b - 8a - 2 + 7abG + 13a^2 G + 2aG \\ B' &= -7abG + 2b - 13a^2 G - 2aG - \frac{2}{G} \end{aligned}$$

for unslotted CARMA. The throughput of unslotted FAMA-NTR is [5],

$$S_{fama} = \frac{1}{b + 1 + \frac{(2 - e^{-aG})}{G} + e^{aG}(4a + b)} \tag{24}$$

For slotted CARMA we obtain

$$S \geq \frac{2aG \cdot e^{-aG} - 2}{A' \cdot e^{-aG} + B'} \tag{25}$$

where

$$\begin{aligned} A' &= b + 4a + 10a^2 G + \frac{2}{G} + 7abG - \frac{2b}{aG} + 2aG \\ B' &= -10a - \frac{4}{G} - 7b + \frac{2b}{aG} - 2 \end{aligned}$$

while the throughput of slotted FAMA-NTR is [5]

$$S_{fama} = \frac{aGe^{-aG}}{(a + b + 1)aG \cdot e^{-aG} + (3a + b)(1 - e^{-aG}) + a} \tag{26}$$

Table IV summarizes the protocol parameters used in our comparison.

Network Speed	Packet Size	δ	$a = \frac{\tau}{\delta}$	$b = \frac{\gamma}{\delta}$
9600 bps	424 bits	44166.7 μs	0.00012	0.377
9600 bps	3200 bits	333333.3 μs	0.0000162	0.050
1 Mbps	424 bits	424 μs	0.0127	0.377
1 Mbps	3200 bits	3200 μs	0.00168	0.050

TABLE IV
PROTOCOL VARIABLES FOR LOW-SPEED NETWORKS (9600 BPS) AND HIGH-SPEED NETWORKS (1 MBPS) WITH TWO TYPES OF DATA PACKETS, SMALL (424 BITS) OR LARGE (3200 BITS). THE CHANNEL DELAY $\tau = 5.4\mu$S, WHILE THE CONTROL PACKETS ARE 160 BITS LONG.

Figs. 2 and 3 show the average throughput (S) versus the offered load (G) for CARMA and FAMA-NTR. It is clear that slotting does not provide much performance improvement in CARMA, and that to achieve high throughput the size of the control packets need to be small compared to the length of the data packets or packet trains. CARMA behaves like FAMA-NTR when the offered load is small. As the offered load increases, the throughput of FAMA-NTR decreases rapidly, while CARMA initially decreases reaching a constant throughput value. This value is obtained by taking $\lim_{\lambda \to \infty} S$. For unslotted CARMA we have

$$\lim_{\lambda \to \infty} S = \lim_{G \to \infty} S = \frac{2}{13a + 7b + 2} = \frac{2\delta}{13\tau + 7\gamma + 2\delta} \tag{27}$$

while for the slotted CARMA

$$\lim_{\lambda \to \infty} S = \lim_{G \to \infty} S = \frac{2}{10a + 7b + 2} = \frac{2\delta}{10\tau + 7\gamma + 2\delta} \tag{28}$$

For slotted as well as for unslotted FAMA-NTR, $\lim_{\lambda \to \infty} S_{fama} = 0$. If we were to have a perfect FAMA protocol in which no collisions of RTSs ever occur, and a constant flow of data packets, the best possible throughput would be

$$S_{max} = \frac{\delta}{3\tau + 2\gamma + \delta} \tag{29}$$

The ratio of S to S_{max} for unslotted CARMA is then

$$1 \geq \frac{S}{S_{max}} \geq \frac{(6\tau + 4\gamma + 2\delta)}{(13\tau + 7\gamma + 2\delta)} \quad (30)$$

while the ratio for slotted CARMA changes slightly to

$$1 \geq \frac{S}{S_{max}} \geq \frac{(6\tau + 4\gamma + 2\delta)}{(10\tau + 7\gamma + 2\delta)} \quad (31)$$

The above result is very encouraging since $\tau < \gamma \ll \delta$. The result also indicates an improvement if the parameter b is small. In practice, this effect can be achieved by allowing a station to transmit longer data packets or multiple packets per floor acquisition.

To verify that the value of S approximated using an infinite population and the upper bounds on average costs for collision resolution times provides a good lower bound for any traffic load, we simulated slotted CARMA using 65 stations that generate RTSs according to a Poisson probability distribution function. The simulations were done ten times for each given $m = \tau\lambda$ value to insure convergence. The results of the simulation are shown in Fig. 3 only for the case of long data packets in a high-speed network, and indicate that our analysis provides a very good approximation of the average throughput.

V. Conclusions

We have described and analyzed a specific protocol, CARMA, as an example of the integration of collision resolution of RTSs in a floor acquisition multiple access protocol. Our analysis shows that collision resolution improves the performance of FAMA protocols substantially. The reason is that the average time it takes for a collision-resolution algorithm to resolve the collisions of RTSs is much smaller than the time used to transmit the associated data packet trains, which are sent with no collisions due to floor acquisition. We have shown that, as the arrival rate of RTSs increases, the throughput achieved by CARMA is close to the maximum throughput that any FAMA protocol can achieve if propagation delays and the control packets used to acquire the floor are much smaller than the data packet trains sent by stations.

Our work continues to address more detailed analysis of performance of this type of protocols, more sophisticated collision-resolution strategies, and the application of CARMA protocols to networks with multiple hops and multiple channels.

Appendix

Proof of Theorem 1: From Table I, we know that $\overline{\mathcal{Z}}(n,0) = 1$, $\overline{\mathcal{Z}}(n,1) = 0$ and that $\overline{\mathcal{Z}}(n,n) = 0$. Let $n = 2$ and $m = 2$; therefore, $\frac{\overline{\mathcal{Z}}(2,2)}{2} = 0 \leq \frac{1}{2}$. Now we assume that, for all $2 \leq n \leq \alpha$ and all $2 \leq m \leq \nu$, the conditions $\overline{\mathcal{Z}}(\alpha, m) \leq \frac{m}{2}$ are satisfied, we show that the condition holds for all $\frac{\overline{\mathcal{Z}}(n,m)}{m}$. Eq. (6) is divided by m and we obtain the general equation

$$\frac{\overline{\mathcal{Z}}(n,m)}{m} = \frac{1}{m}\sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \left[\overline{\mathcal{Z}}(\alpha, m-i) + \overline{\mathcal{Z}}(\beta, i)\right] \quad (32)$$

There are three cases to consider according to the summation indices.

Case 1: $m \leq \alpha$ and $m \leq \beta$: Then $\mu = 0$ while $\nu = m$. Therefore,

$$\frac{\overline{\mathcal{Z}}(n,m)}{m} = \frac{1}{m}\sum_{i=0}^{m} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} \left[\overline{\mathcal{Z}}(\alpha, m-i) + \overline{\mathcal{Z}}(\beta, i)\right] \quad (33)$$

Extracting the first two and the last two elements from the summation (i.e., the elements with $i = 0, 1, m-1, m$) and noting that $\overline{\mathcal{Z}}(\alpha,1) = \overline{\mathcal{Z}}(\beta,1) = 0$, and $\overline{\mathcal{Z}}(\alpha,0) = \overline{\mathcal{Z}}(\beta,0) = 1$, we obtain,

$$\begin{aligned}\frac{\overline{\mathcal{Z}}(n,m)}{m} &= \sum_{i=2}^{m-1} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{m\binom{n}{m}} \left[\overline{\mathcal{Z}}(\alpha, m-i) + \overline{\mathcal{Z}}(\beta, i)\right] \\ &+ \frac{\binom{\alpha}{m}\binom{\beta}{0}}{m\binom{n}{m}}\left[\overline{\mathcal{Z}}(\alpha, m) + 1\right] + \frac{\binom{\alpha}{m-1}\binom{\beta}{1}}{m\binom{n}{m}}\overline{\mathcal{Z}}(\alpha, m-1) \\ &+ \frac{\binom{\alpha}{1}\binom{\beta}{m-1}}{m\binom{n}{m}}\overline{\mathcal{Z}}(\beta, m-1) + \frac{\binom{\alpha}{0}\binom{\beta}{m}}{m\binom{n}{m}}\left[1 + \overline{\mathcal{Z}}(\beta, m)\right]\end{aligned} \quad (34)$$

Because $\overline{\mathcal{Z}}(\alpha, m) \leq \frac{m}{2}$ and $\overline{\mathcal{Z}}(\beta, m) \leq \frac{m}{2}$, we obtain

$$\begin{aligned}\frac{\overline{\mathcal{Z}}(n,m)}{m} &\leq \frac{1}{2}\sum_{i=0}^{m} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} + \frac{2}{2m}\frac{\binom{\alpha}{m}\binom{\beta}{0}}{\binom{n}{m}} \\ &- \frac{1}{2m}\frac{\binom{\alpha}{m-1}\binom{\beta}{1}}{\binom{n}{m}} - \frac{1}{2m}\frac{\binom{\alpha}{1}\binom{\beta}{m-1}}{\binom{n}{m}} \\ &+ \frac{2}{2m}\frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}}\end{aligned} \quad (35)$$

For any binomial coefficient, $\binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1}$. Therefore, noticing that the sum from Eq. (35) equals one, we have

$$\begin{aligned}\frac{\overline{\mathcal{Z}}(n,m)}{m} &\leq \frac{1}{2} + \frac{2\alpha - m\beta - 2m + 2}{2m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}} \\ &+ \frac{2\beta - m\alpha - 2m + 2}{2m^2}\frac{\binom{\beta}{m-1}}{\binom{n}{m}}\end{aligned} \quad (36)$$

For the equation $\frac{\overline{\mathcal{Z}}(n,m)}{m} \leq \frac{1}{2}$ to be true, the last two terms in Eq. (36) must be zero or negative. If n is even, then $\alpha = \beta$ and

$$\frac{\overline{\mathcal{Z}}(n,m)}{m} \leq \frac{1}{2} + \frac{\binom{\alpha}{m-1}}{m^2\binom{n}{m}}\left[\alpha(2-m) + (2-2m)\right] \quad (37)$$

Because $m > 1$, $\alpha(2-m)+(2-2m) < 0$; therefore, $\frac{\overline{\mathcal{Z}}(n,m)}{m} \leq \frac{1}{2}$. If n is odd, then we have $\beta = \alpha - 1$; accordingly, we have

$$\begin{aligned}\frac{\overline{\mathcal{Z}}(n,m)}{m} &\leq \frac{1}{2} + \frac{\alpha(2-m) + (2-m)}{2m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}} \\ &+ \frac{\alpha(2-m) - 2m}{2m^2}\frac{\binom{\beta}{m-1}}{\binom{n}{m}}\end{aligned} \quad (38)$$

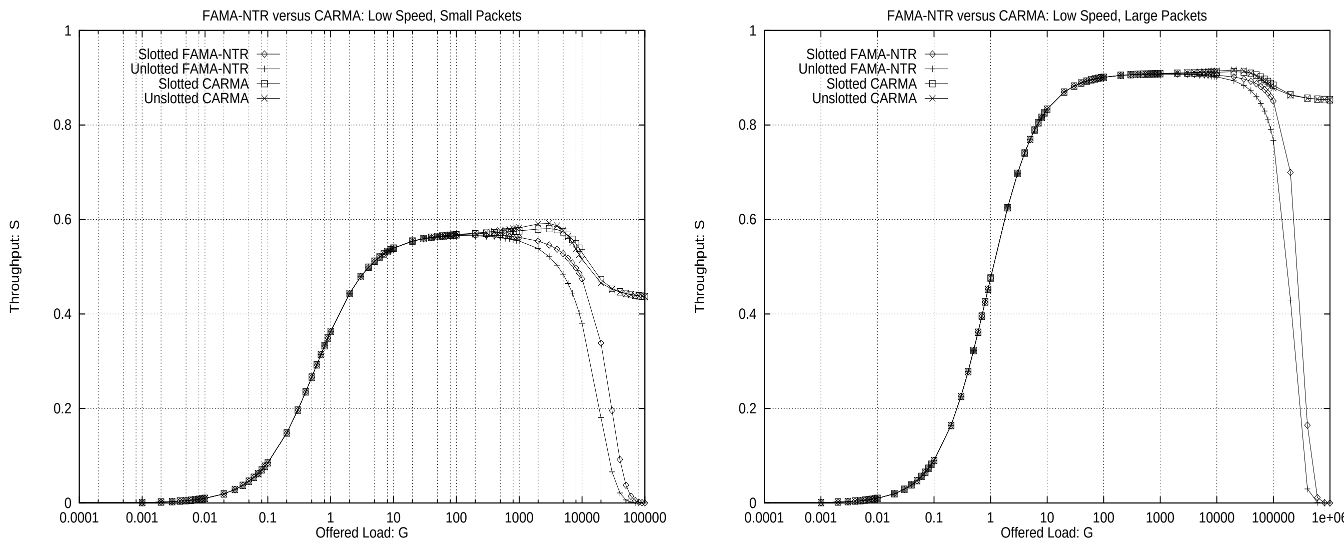

Fig. 2. Throughput of FAMA-NTR and CARMA for low-speed network.

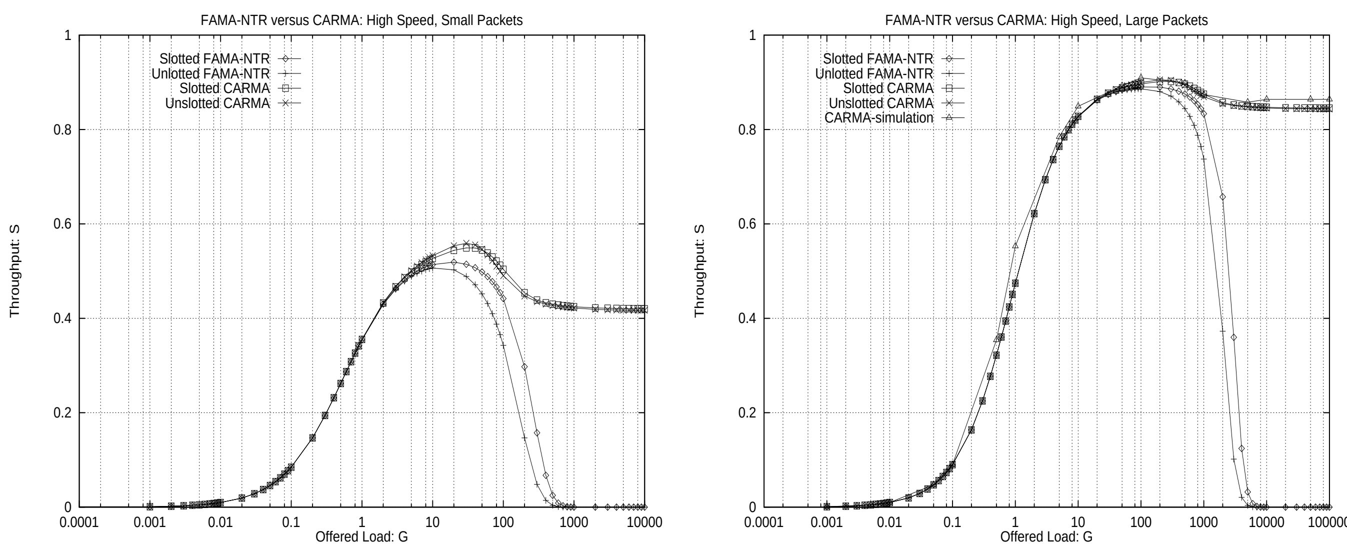

Fig. 3. Throughput of FAMA-NTR and CARMA for high-speed network.

Because $m > 1$, $(\alpha(2-m)+(2-m)) \leq 0$ and $(\alpha(2-m)-2m) \leq 0$; therefore, our assumption that $\overline{\mathcal{Z}}(n,m) \leq \frac{m}{2}$ is correct for any n and any $m > 1$.

Case 2: $m \leq \alpha$ and $m > \beta$: Then $\mu = 0$ while $\nu = \beta$ and n can only be odd. Further more, $\alpha = m$ and $\beta = m-1$, while $n = 2m-1$. Substituting all this in Eq. (32) we obtain

$$\frac{\overline{\mathcal{Z}}(n,m)}{m} = \sum_{i=2}^{m-2} \frac{\binom{m}{m-i}\binom{m-1}{i}}{m\binom{2m-1}{m}}\left[\mathcal{Z}(m,m-i)+\mathcal{Z}(m-1,i)\right] + \frac{\binom{m}{m}\binom{m-1}{0}}{m\binom{n}{m}} + \frac{\binom{m}{m-1}\binom{m-1}{1}}{m\binom{n}{m}}\mathcal{Z}(m,m-1) \quad (39)$$

where several terms have been evaluated following the rules in Table I. Because $\mathcal{Z}(m,m-i) \leq \frac{m-i}{2}$ and $\mathcal{Z}(m-1,i) \leq \frac{i}{2}$, we obtain

$$\frac{\overline{\mathcal{Z}}(n,m)}{m} = \frac{1}{2} + \frac{2-2m^2}{2m}\frac{1}{\binom{n}{m}} \leq \frac{1}{2} \quad (40)$$

We conclude that our assumption $\overline{\mathcal{Z}}(n,m) \leq \frac{m}{2}$ is correct for any n and all $m > 1$.

Case 3: $m > \alpha$ and $m > \beta$: Then $\mu = m - \alpha$ while $\nu = \beta$. Once again, because $\mathcal{Z}(\alpha,i) \leq \frac{i}{2}$ and $\mathcal{Z}(\beta,i) \leq \frac{i}{2}$, using the same procedure as for Cases 1 and 2, we obtain

$$\frac{\overline{\mathcal{Z}}(n,m)}{m} = \frac{1}{2} - \frac{\beta}{2m}\frac{\binom{\alpha}{m-\beta}\binom{\beta}{m}}{\binom{n}{m}} \quad (41)$$

We conclude that our assumption $\overline{\mathcal{Z}}(n,m) \leq \frac{m}{2}$ is correct for any n and all $m > 1$. Therefore, for all $m > 2$ and any value of n, $\frac{\overline{\mathcal{Z}}(n,m)}{m} \leq \frac{1}{2}$, and $\overline{\mathcal{Z}}(n,m) \leq \frac{m}{2}$. □

Proof of Theorem 2: According to Eq. (3), the ratio

$\frac{\overline{\mathcal{C}}(n,m)}{m}$ can be expressed as

$$\frac{\overline{\mathcal{C}}(n,m)}{m} = \frac{1}{m}\sum_{i=\mu}^{\nu}\frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}}\left[\mathcal{C}(\alpha, m-i)+\mathcal{C}(\beta,i)+1\right] \tag{42}$$

Case 1: $m \leq \alpha$ and $m \leq \beta$: Then $\mu = 0$ while $\nu = m$. Let us separate the first two and the last two terms from the summation and evaluate them, our expression becomes

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &= \frac{1}{m}\sum_{i=2}^{m-2}\frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}}\left[\mathcal{C}(\alpha, m-i)+\mathcal{C}(\beta,i)+1\right]\\ &+ \frac{1}{m}\frac{\binom{\alpha}{m}\binom{\beta}{0}}{\binom{n}{m}}\left[\mathcal{C}(\alpha,m)+1\right]\\ &+ \frac{1}{m}\frac{\binom{\alpha}{m-1}\binom{\beta}{1}}{\binom{n}{m}}\left[\mathcal{C}(\alpha,m-1)+1\right]\\ &+ \frac{1}{m}\frac{\binom{\alpha}{1}\binom{\beta}{m-1}}{\binom{n}{m}}\left[\mathcal{C}(\beta,m-1)+1\right]\\ &+ \frac{1}{m}\frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}}\left[\mathcal{C}(\beta,m)+1\right]\end{aligned} \tag{43}$$

Following the procedure introduced in the proof of Theorem 1, we can substitute $\mathcal{C}(\beta,i)$ and $\mathcal{C}(\alpha,i)$ by $\frac{3i}{2}-1$. We then proceed to collect the missing terms from the summation and arrive at the following expression:

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &\leq \frac{3}{2}-\frac{1}{m}+\frac{2}{2m}\frac{\binom{\alpha}{m}\binom{\beta}{0}}{\binom{n}{m}}-\frac{1}{2m}\frac{\binom{\alpha}{m-1}\binom{\beta}{1}}{\binom{n}{m}}\\ &- \frac{1}{2m}\frac{\binom{\alpha}{1}\binom{\beta}{m-1}}{\binom{n}{m}}+\frac{2}{2m}\frac{\binom{\alpha}{0}\binom{\beta}{m}}{\binom{n}{m}}\end{aligned} \tag{44}$$

Which can be simplified using the binomial coefficient identity introduced in Theorem 1 to

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &\leq \frac{3}{2}-\frac{1}{m}+\frac{2\alpha-2m+2-\beta m}{2m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}}\\ &+ \frac{2\beta-2m+2-\alpha m}{2m^2}\frac{\binom{\beta}{m-1}}{\binom{n}{m}}\end{aligned} \tag{45}$$

Assume that n is even, then $\alpha = \beta = \frac{n}{2}$ and

$$\frac{\overline{\mathcal{C}}(n,m)}{m} \leq \frac{3}{2}-\frac{1}{m}+\frac{\alpha(2-m)+2(1-m)}{m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}} \leq \frac{3}{2}-\frac{1}{m} \tag{46}$$

On the other hand, if n is odd, then $\beta = \alpha - 1$, and

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &\leq \frac{3}{2}-\frac{1}{m}+\frac{\alpha(2-m)+(2-m)}{2m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}}\\ &+ \frac{\alpha(2-m)-2m}{2m^2}\frac{\binom{\alpha-1}{m-1}}{\binom{n}{m}} \leq \frac{3}{2}-\frac{1}{m}\end{aligned} \tag{47}$$

Therefore, $\overline{\mathcal{C}}(n,m) \leq \frac{3m}{2}-1$ for any n and all $m > 1$.

Case 2: $m \leq \alpha$ and $m > \beta$: Then $\mu = 0$ while $\nu = \beta$ and n can only be odd. Further more, $\alpha = m$ and $\beta = m-1$, while $n = 2m-1$. Substituting all this in Eq. (3), we obtain

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &= \sum_{i=2}^{\beta-2}\frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{m\binom{n}{m}}\left[\mathcal{C}(\alpha,m-i)+\mathcal{C}(\beta,i)+1\right]\\ &+ \frac{\binom{\alpha}{m}\binom{\beta}{0}}{m\binom{n}{m}}\left[\mathcal{C}(\alpha,m)+1\right]\\ &+ \frac{\binom{\alpha}{m-1}\binom{\beta}{1}}{m\binom{n}{m}}\left[\mathcal{C}(\alpha,m-1)+1\right]\\ &+ \frac{\binom{\alpha}{m-\beta+1}\binom{\beta}{\beta-1}}{m\binom{n}{m}}\left[\mathcal{C}(\alpha,m-\beta+1)+\mathcal{C}(\beta,\beta-1)+1\right]\\ &+ \frac{\binom{\alpha}{m-\beta}\binom{\beta}{\beta}}{m\binom{n}{m}}\left[\mathcal{C}(\alpha,m-\beta)+\beta\right]\end{aligned} \tag{48}$$

From our induction assumption, for all α and $\beta < n$, $\mathcal{C}(\alpha,i) \leq \frac{3i}{2}-1$ and $\mathcal{C}(\beta,i) \leq \frac{3i}{2}-1$. Thus,

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &\leq \frac{3}{2}-\frac{1}{m}+\frac{2}{2m}\frac{\binom{\alpha}{m}}{\binom{n}{m}}\\ &- \frac{\beta}{2m}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}}-\frac{\beta}{2m}\frac{\binom{\alpha}{m-\beta}}{\binom{n}{m}}\end{aligned} \tag{49}$$

From the binomial coefficient property brought up in Theorem 1, we can simplify the above equation to

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} &\leq \frac{3}{2}-\frac{1}{m}+\frac{2\alpha-m\beta-2m+2}{2m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}}\\ &- \frac{\beta}{2m}\frac{\binom{\alpha}{m-\beta}}{\binom{n}{m}}\end{aligned} \tag{50}$$

Because $m \leq \alpha$ and $m > \beta$, n can only be odd and $\alpha = m$, while $\beta = \alpha - 1$. Accordingly, we have

$$\begin{aligned}\frac{\overline{\mathcal{C}}(n,m)}{m} \leq \frac{3}{2} &-\frac{1}{m}+\frac{\alpha(2-m)+(2-m)}{2m^2}\frac{\binom{\alpha}{m-1}}{\binom{n}{m}}\\ &- \frac{\beta}{2m}\frac{\binom{\alpha}{m-\beta}}{\binom{n}{m}} \leq \frac{3}{2}-\frac{1}{m}\end{aligned} \tag{51}$$

We conclude that, for any n and all $m > 1$, $\overline{\mathcal{C}}(n,m) \leq \frac{3m}{2}-1$ is correct.

Case 3: $m > \alpha$ and $m > \beta$: Then $\mu = m - \alpha$ while $\nu = \beta$. Therefore, substituting all this in Eq. (3),

$$\frac{\overline{\mathcal{C}}(n,m)}{m} = \frac{1}{m}\sum_{i=m-\alpha}^{\beta}\frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}}\left[\mathcal{C}(\alpha,m-i)+\mathcal{C}(\beta,i)+1\right] \tag{52}$$

Notice that terms with $i = 0$ do not appear in the equation. The smallest value for i is 1. Because $\mathcal{C}(\beta,1) = \mathcal{C}(\alpha,1) = 0$, it is true that $\mathcal{C}(\beta,1) = \mathcal{C}(\alpha,1) \leq \frac{3}{2}-1$; therefore,

$$
\begin{aligned}
\frac{\overline{\mathcal{C}}(n,m)}{m} &\leq \sum_{i=m-\alpha}^{\beta} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{m\binom{n}{m}}\left[\frac{3(m-i)}{2}-1+\frac{3i}{2}-1+1\right] \\
&= \left(\frac{3}{2}-\frac{1}{m}\right)\sum_{i=m-\alpha}^{\beta} \frac{\binom{\alpha}{m-i}\binom{\beta}{i}}{\binom{n}{m}} = \frac{3}{2}-\frac{1}{m} \qquad (53)
\end{aligned}
$$

We conclude that $\overline{\mathcal{Z}}(n,m) \leq \frac{m}{2}$ is correct for any n and all $m > 1$. We have shown that for all $m > 1$ and for any n, $\frac{\overline{\mathcal{C}}(n,m)}{m} \leq \frac{3}{2} - \frac{1}{m}$; therefore, $\overline{\mathcal{Z}}(n,m) \leq \frac{3m}{2} - 1$. □

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, Second Edition, Prentice-Hall, 1992.

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *Proc. of ACM SIGCOMM '94*, pp. 212–25, ACM, 1994.

[3] K. Biba, "A Hybrid Wireless MAC Protocol Supporting Asynchronous and Synchronous MSDU Delivery Services," Tech. Rep. Paper 802.11/91-92, IEEE 802.11 Working Group, 1992.

[4] R. L. Brewster and A. M. Glass, "Throughput Analysis of Non-Persistent and Slotted Non-Persistent CSMA/CA Protocols," in *Proc. 4th International Conference on Land Mobile Radio*, pp. 231–6, Institution of Electronic and Radio Engineers, 1987.

[5] C. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access for Packet-Radio Networks," *Proc. ACM SIGCOMM 95*, Cambridge, MA, August 30-September 1, 1995.

[6] C. Fullmer and J.J. Garcia-Luna-Aceves, "FAMA-PJ: A Channel Access Protocol for Wireless LANs," *Proc. ACM Mobile Computing and Networking '95*, Berkeley, CA, Nov. 14-15, 1995.

[7] P. Karn, "MACA - a new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–40, ARRL, 1990.

[8] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[9] W. F. Lo and H. T. Mouftah, "Carrier Sense Multiple Access with Collision Detection for Radio Channels," in *IEEE 13th International Communications and Energy Conference*, pp. 244–47, IEEE, 1984.

[10] G. S. Sidhu, R. F. Andrews, and A. B. Oppenheimer, *Inside AppleTalk, Second Edition*. Addison-Wesley Publishing Company, Inc., 1990.

[11] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1417–1433, 1975.